DESIGNING AND VERIFYING AUTONOMOUS VEHICLE PERCEPTION SYSTEMS

Jelena Pisarov
Nikola Tesla - Serbian Cultural and Educational Center, Budapest, Hungary
jelena.pisarov@gmail.com

Abstract: *Engineers have long used Matlab to design and verify perception systems for autonomous vehicles. One of the primary challenges in this domain is handling large volumes of heterogeneous sensor data, which requires effective tools for visualisation and interpretation. This work highlights the role of Application Programming Interfaces (APIs) that enable engineers to plot, analyse, and understand diverse datasets collected from autonomous driving environments. A second key aspect is object detection in images, supported by computer vision toolboxes and accelerated through automated ground-truth labelling workflows. Such labeled data is essential for developing new object detectors and validating existing ones. A further challenge arises from combining detections from multiple sensors, where discrepancies between vision-based and radar-based detections must be resolved through tracking and sensor fusion. Matlab provides multi-object tracking capabilities and synthetic detection generation, enabling comprehensive testing and verification of perception algorithms. Overall, the tools discussed support a streamlined workflow for designing, evaluating, and improving autonomous vehicle perception systems.*

Key Words: Matlab, autonomous driving, perception systems, object detection, sensor fusion

1. INTRODUCTION

Perception systems represent a fundamental component of autonomous vehicle technology, enabling vehicles to interpret their surroundings and make informed driving decisions. These systems rely on a diverse set of sensors – such as cameras, radar, lidar, and inertial measurement units – to capture complementary information about the environment. The complexity and heterogeneity of the collected data require robust tools for visualisation, analysis, and algorithm development. Matlab has emerged as a widely adopted platform for this purpose, offering integrated toolboxes that support data exploration, object detection, ground-truth labelling, multi-sensor fusion, and multi-object tracking [1-5].

Developing reliable perception algorithms involves several key challenges. Engineers must first understand and interpret large volumes of raw sensor data, which often come in different formats and coordinate systems. They must also design and validate object detection models capable of identifying vehicles, pedestrians, lanes, and other relevant features. Furthermore, autonomous systems must reconcile conflicting or incomplete detections from multiple sensors, requiring advanced fusion and tracking techniques to produce accurate and stable estimates of the environment.

This paper examines how Matlab's perception-related tools support these tasks, with particular emphasis on data visualisation, object detection workflows, ground-truth labelling, and multi-sensor fusion. The following section provides an overview of vehicle sensor data and demonstrates how visualisation APIs can be used to interpret and analyze logged data from automated driving applications.

2. VISUALISING IN IMAGE AND VEHICLE COORDINATES

Effective visualisation is a key component in understanding how perception systems interpret sensor data in automated driving applications. Matlab provides a set of tools that allow engineers to inspect detections both in image coordinates – directly on the video frame – and in vehicle-centric ground-plane coordinates. Combining these two perspectives enables a comprehensive understanding of how sensors perceive the environment and how detection algorithms operate [6-20].

2.1 Visualising in Image Coordinates

The visualisation process typically begins by selecting a specific timestamp in the recorded dataset and extracting the corresponding video frame. Using classic image-processing functions such as *imshow*, the frame is displayed in image coordinates, forming a basis for overlaying additional sensor information. This approach allows engineers to directly compare raw camera imagery with the outputs of perception algorithms.

As shown in Figure 1, Matlab enables the extraction

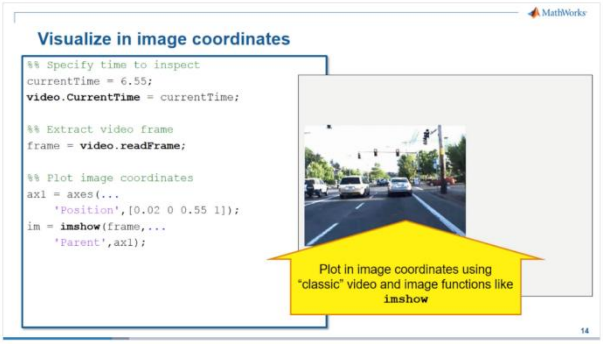

Figure 1 Visualisation of a video frame in image coordinates using standard Matlab image-processing functions.

and display of individual video frames, which serve as the foundation for further visualisation of detections and sensor outputs [20-25].

2.2 Visualising Detections in Vehicle Coordinates

After visualising the scene in image coordinates, the next step is to examine detections in vehicle-centric coordinates. Matlab provides specialized APIs for this purpose, beginning with the *coverageAreaPlotter*, which displays the sensor's field of view and operational range. The API is divided into an initialisation step and an update step, ensuring efficient rendering during playback or iterative analysis.

Once the coverage area is established, detections can be plotted using the *detectionPlotter*. This tool supports visualisation of object positions, classification labels, and relative velocity vectors. The position is typically represented by a marker, the label indicates the detected object class, and the velocity vector illustrates relative motion between the ego vehicle and the detected object.

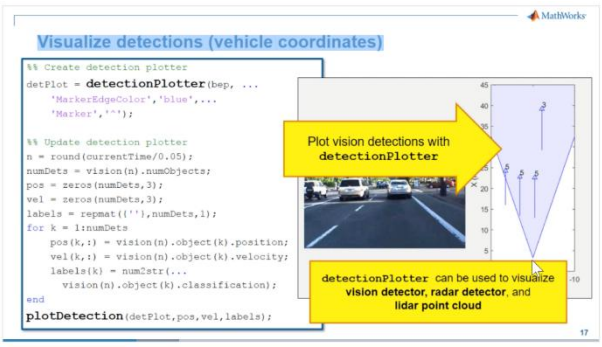

Figure 2 Visualisation of object detections in vehicle coordinates using the Matlab detectionPlotter API.

This process is illustrated in Figure 2, which shows how detections from a vision-based detector are plotted in vehicle coordinates. The same API can be used for radar and lidar detections, enabling consistent visualisation across different sensor modalities [26-33].

3. DETECTING OBJECTS IN IMAGES

Object detection represents one of the core capabilities of perception systems in automated driving applications. The goal of an object detector is to identify relevant objects within an image – such as vehicles, pedestrians, cyclists, or traffic signs – and to return both their classification labels and corresponding bounding boxes. This process is illustrated in Figure 3, which shows the relationship between training data, ground-truth annotations, and the resulting detector.

Developing an object detector requires reversing the typical perception workflow. Instead of starting with a trained model and applying it to images, engineers must begin with raw images and manually annotated ground-truth data. Ground truth consists of bounding boxes and classification labels that describe the true positions and identities of objects within each image. These annotations serve as the foundation for training machine learning or deep learning–based detection models.

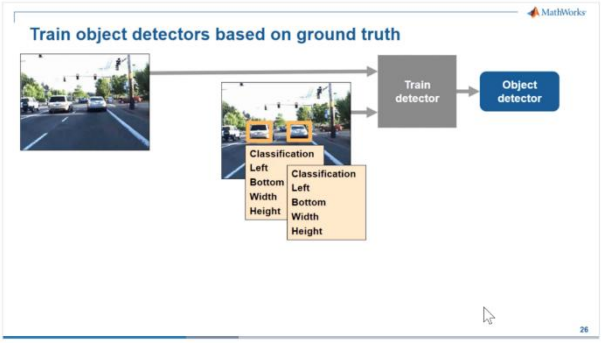

Figure 3 Training object detectors based on ground-truth annotations.

To construct a high-quality training dataset, engineers must annotate each image with precise object locations and classes. In automated driving applications, these images are rarely isolated; they typically originate from video sequences. This means that consecutive frames are temporally related, and annotation tools must support efficient labelling across sequences rather than individual static images. Matlab provides a ground-truth labelling app designed specifically for this purpose, enabling users to label objects across video frames and significantly accelerate the annotation process.

Once images and ground-truth annotations are prepared, Matlab's *Computer Vision Toolbox* and *Deep Learning Toolbox* offer a unified interface for training a variety of object detectors. These include classical machine learning approaches as well as modern deep learning architectures, which have become increasingly prevalent in automated driving systems. Because all training functions share a consistent interface, engineers can easily experiment with different detector types and compare their performance using the same dataset.

Ground-truth data is not only essential for training new detectors but also plays a critical role in evaluating existing ones. By comparing detector outputs with ground-truth annotations, engineers can measure accuracy, assess failure cases, and refine detection algorithms. This dual role – supporting both training and evaluation – makes ground-truth labelling a central component of the perception development workflow.

4. FUSING MULTIPLE VEHICLE DETECTIONS

In automated driving applications, it is common for multiple detection algorithms – often based on different sensor modalities – to return slightly different estimates of an object's position. Engineers must therefore reconcile

these discrepancies to obtain a more accurate and stable understanding of the environment. This is achieved through **sensor fusion** and **multi-object tracking** techniques.

Consider a scenario in which a vision-based object detector estimates the position of a nearby vehicle. At the same time, a radar sensor on the same ego vehicle produces its own detection, placing the object in a slightly different location. The goal of fusion is to combine these independent measurements to produce a single, improved estimate of the object's true position.

To accomplish this, detections from different sensors are fed into a **multi-object tracker**, which integrates information over time and across modalities. As shown in Figure 4, the tracker produces object tracks – represented as black bounding boxes – that may follow radar detections, vision detections, or a fused combination of both.

When radar and vision detections are spatially close, the tracker fuses them into a single track. This fused track typically provides a more accurate estimate of the object's position than either sensor alone. When the fused tracks are overlaid on lidar data, the improvement becomes even more apparent: the fused estimate aligns more closely with the true object location.

Figure 4 Results of a multi-object tracker combining radar and vision detections.

Fusion and tracking are not only useful for perception – they also enable downstream algorithms. In this example, the output of the multi-object tracker is used to support a **forward collision warning** system. Matlab's implementation of this algorithm supports C code generation, allowing engineers to transition from simulation to prototyping in embedded applications.

While the previous example relied on real sensor data collected from a vehicle, there are situations where engineers need to design or test fusion algorithms without access to real-world data. For such cases, Matlab supports **scenario-based simulation**, where an ego vehicle (typically shown as a blue car) is equipped with simulated vision and radar sensors. Other vehicles – such as the orange vehicle in the example – can be scripted to move through different sensor coverage areas, generating synthetic detections for testing fusion logic.

As illustrated in Figure 5, the orange vehicle is not consistently detected by any single sensor as it passes the ego vehicle on the left. However, the multi-object tracker fuses detections from all available sensors, producing a continuous and reliable track even when individual sensors temporarily lose sight of the object.

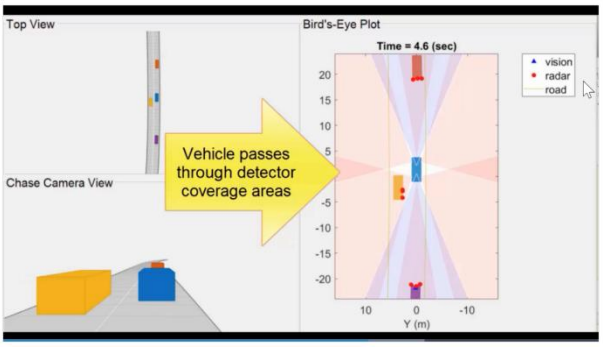

Figure 5 Fusion of multiple sensors provides consistent tracking of a passing vehicle.

The multi-object tracker provided in the Automated Driving Toolbox accepts detections along with their timestamps, measurement values, and noise characteristics. Its outputs include estimated object states, state covariances, and track identifiers. Conceptually, the tracker consists of two main components:

1. **Tracking Filter** – predicts and updates the state of each track. It may be configured as a linear, extended, or unscented Kalman filter.
2. **Track Manager** – assigns detections to tracks. It determines whether a detection corresponds to an existing track, should initialize a new track, or whether an existing track should be terminated due to missing detections.

To simplify tracker configuration, Matlab provides several initialisation functions, including constant-velocity, constant-acceleration, and constant-turn models. When instantiating the multi-object tracker, engineers specify the desired initialisation function (e.g., *initcaekf* for an extended Kalman filter with a constant-acceleration model) along with parameters for the Track Manager.

In fusion scenarios, detections from different sensors – such as radar and vision – must be mapped into a common measurement format before being passed to the tracker. This typically involves converting sensor-specific outputs (e.g., position, velocity, time) into a unified measurement vector that the tracker can process. Once this mapping is complete, the multi-object tracker can seamlessly fuse detections from multiple sources, producing robust and consistent object tracks.

5. CONCLUSIONS

Autonomous vehicle perception systems rely on the integration of sensor data processing, object detection, ground-truth generation, and multi-sensor fusion. Matlab provides a comprehensive ecosystem that supports each stage of this workflow, enabling engineers to design, visualise, and validate perception algorithms efficiently. Through

tools for image-based and vehicle-centric visualisation, developers gain insight into how different sensors interpret the driving environment. Ground-truth labelling applications streamline the creation of annotated datasets, which are essential for training and evaluating machine-learning and deep-learning-based object detectors. Multi-object tracking and sensor fusion techniques further enhance perception robustness by reconciling discrepancies between heterogeneous sensor outputs and producing stable, accurate object trajectories.

The examples presented in this work demonstrate how Matlab's integrated toolboxes facilitate both data-driven development and scenario-based simulation, supporting rapid prototyping and deployment. By combining visualisation, detection, labelling, fusion, and tracking capabilities within a unified framework, Matlab enables a streamlined and scalable approach to designing and verifying autonomous vehicle perception systems. As automated driving technologies continue to evolve, such tools will remain essential for ensuring reliability, safety, and performance in increasingly complex environments.

6. REFERENCES

[1] MTMT, https://m2.mtmt.hu/, last accessed 2026/02/20.

[2] Ugljesa Marjanovic, Gyula Mester and Bojana Milic Marjanovic, *Assessing the Success of Artificial Intelligence Tools: an Evaluation of ChatGPT Using the Information System Success Model*, Interdisciplinary Description of Complex Systems, Indecs, Vol. 22, Issue 3, pp. 266-275, DOI: 10.7906/indecs.22.3.3, 2024.

[3] Dalma Zilahy, Gyula Mester, *Factors Affecting the Adoption of Self-Driving Cars*, Interdisciplinary Description of Complex Systems, Indecs, Vol. 22, Issue 6, pp. 732-737, DOI: 10.7906/indecs.22.6.7, 2024.

[4] János Sánta, *Artificial Intelligence Applied in Enterprise Resource Planning*, Interdisciplinary Description of Complex Systems: Indecs, Vol. 22, Issue 3, pp. 360-363, DOI: 10.7906/indecs.22.3.11, ISSN 1334-4684, 2024.

[5] César Bautista, Gyula Mester, *Internet of Things in Self-Driving Cars Environment*, Interdisciplinary Description of Complex Systems, Indecs, Vol. 21, Issue 2, pp. 188-198, DOI: 10.7906/indecs.21.2.8, ISSN 1334-4684, 2023.

[6] Dalma Zilahy, Gyula Mester, *Managing Negative Emotions Caused by Self-Driving*, Interdisciplinary Description of Complex Systems, Indecs, Vol. 21, Issue 4, pp. 351-355, DOI: 10.7906/indecs.21.4.4, 2023.

[7] Jelena Pisarov, Gyula Mester, *Self-Driving Robotic Cars: Cyber Security Developments*, Research Anthology on Cross-Disciplinary Designs and Applications of Automation, IGI Global, pp. 969-1001, 2022.

[8] Gyula Mester, *The 2022 Ranking List of Citation Analysis Researchers using h-Index*, Interdisciplinary Descrip-tion of Complex Systems: Indecs, Vol. 20, Issue 6, pp. 775-779, ISSN 1334-4684, 2022.

[9] César Bautista, Gyula Mester, *Safety Analysis in Automotive Perception*, Full Texts Book of the EJONS 13th Int-ernational Conference on Mathematics, Engineering, Natural & Medical Science, Edited by Dr. Mehriban Emek, Nurlan Akhmetov, pp. 368-378, ISBN: 978-625-7464-40-6, Cappadocia, Turkey, October 26-27, 2021.

[10] Gyula Mester, César Bautista, *Automotive Digital Perception*, Review of the National Center for Digitization, ISSN:1820-0109, Issue: 39, pp. 90-95, 2021.

[11] Gyula Mester, Jelena Pisarov *Academic Ranking of World Universities 2021*, Review of the National Center for Digitization, ISSN: 1820-0109, Issue: 39, pp. 83-89, 2021.

[12] Gyula Mester, Jelena Pisarov, *Digitalization in Modern Transport of Passengers and Freight*, Review of the National Center for Digitization, ISSN: 1820-0109, Issue: 39, pp. 96-101, 2021.

[13] Jelena L. Pisarov, Gyula Mester, *The use of Autonomous Vehicles in Transportation*, Tehnika, Vol. 76, Issue 2, pp. 171-177, DOI: 10.5937/tehnika2102171P, 2021.

[14] Jelena Pisarov, Gyula Mester, *Implementing New Mobility Concepts with Autonomous Self-Driving Robotic Cars*, IPSI BgD Transactions on Advanced Research (TAR), Vol. 17, Issue 2, ISSN 1820-4511, pp. 41-49, July 2021.

[15] Jelena Pisarov, Gyula Mester, *The Impact of 5G Technology on Life in the 21st Century*, IPSI BgD Transactions on Advanced Research (TAR), Vol. 16, Issue 2, pp. 11-14, ISSN 1820-4511, July 2020.

[16] Jelena Pisarov, Gyula Mester, *The Future of Autonomous Vehicles*, FME Transactions, Vol. 49, No. 1, DOI: 10. 5937/fme2101029P, pp. 29-35, December 2020.

[17] Jelena Pisarov, Gyula Mester, *Rang lista fizičara Srbije*, Proceedings of the XXVI Skup Trendovi razvoja: Inova-cije u modernom obrazovanju, TREND 2020, Kopaonik, Serbia, pp. 559-562, 2020.02.16.

[18] Damir Šostarić, Gyula Mester, *Drone Localization using Ultrasonic TDOA and RSS signal: Integration of the Inverse Method of a Particle Filter*, FME Transactions, Vol. 48, No. 1, pp. 21-30, ISSN 1451-2092, 2020.

[19] Gyula Mester, Jelena Pisarov and Dalma Zilahy, *Magyarországi robotikai kutatók ranglistája*, XXXV. Jubileu-mi Kandó Konferencia 2019 (JKK2019), Budapest, Hungary, pp. 224-233, 2019.11.14-15.

[20] Jelena Pisarov, Gyula Mester, *Programming the mBot Robot in School*, Proceedings of the International Confe-rence & Workshop Mechatronics in Practice and Education, MechEdu 2019, pp. 45-48, ISBN 978-86-918815-5-9, Subotica, Serbia, 12.12.2019.

[21] Gyula Mester, Jelena Pisarov and Endre Németh, *Óbudai Egyetem rangsorolása a Webometrics 2019-es ranglis-tákon*, XXXV. Jubileumi Kandó Konferencia 2019 (JKK2019), Budapest, Hungary, pp. 234-240, 14-15.11.2019.

[22] Janos Simon, Gyula Mester, *Critical Overview of the Cloud-Based Internet of Things Pilot Platforms for Smart*

Cities, Interdisciplinary Description of Complex Systems, Vol. 16, No. 3-A, pp. 397-407, 2018.
[23] Attila Nemes, Gyula Mester, Tibor Mester, *A Soft Computing Method for Efficient Modelling of Smart Cities Noise Pollution*, Interdisciplinary Description of Complex Systems, Indecs, Vol. 16, No. 3-A, pp. 302-312, ISSN 1334-4684, DOI: 10.7906/ indecs.16.3.1, Croatian Interdisciplinary Society, 2018.
[24] Attila Nemes, Gyula Mester, *Unconstrained Evolutionary and Gradient Descent-based Tuning of Fuzzy-Partiti-ons for UAV Dynamic Modeling*, FME Transactions, Vol. 45, Issue 1, pp. 1-8, 2017
[25] Gyula Mester, *Academic Ranking of World Universities*, Review of the National Center for Digitization, Issue 31, pp. 40-45, Faculty of Mathematics, University of Belgrade, 2017.
[26] Gyula Mester, *Rankings Scientists, Journals and Countries Using h-index*, Interdisciplinary Description of Com-plex Systems, Zagreb, Croatian Interdisciplinary Society, Vol. 14, No. 1, pp. 1-9, ISSN 1334-4684, DOI: 10.7906/indecs.14.1.1, 2016.
[27] Gyula Mester, *Design of the Fuzzy Control Systems Based on Genetic Algorithm for Intelligent Robots*, Interdis-ciplinary Description of Complex Systems, Indecs, Vol. 12, Issue 3, pp. 245-254, 2014.
[28] Gyula Mester, *Új tudományos eredmények mérése*, XXX Kandó Conference, Budapest, Hungary, pp. 1-10, ISBN 978-615-5460-24-1, 2014.11.20.
[29] Gyula Mester, *Univerziteti regiona na Šangajskoj rang listi univerziteta u svetu 2012*, Zbornik radova XIX SkupaTrendovi razvoja, pp. 1-5, Kopaonik, Serbia, 2013.
[30] Gyula Mester, Aleksandar Rodic, *Modeling and Navigation of an Autonomous Quad-Rotor Helicopter*, E society Journal: Research and Applications, Vol. 3, No. 1, pp. 45-53, 2012.
[31] Gyula Mester, *The Evaluation of the Impact Factor of the Journal Acta Polytechnica Hungarica*, Proceedings of the TREND Conference, pp. 70-73, 2011.
[32] Gyula Mester, *Felsőoktatási világranglisták 2011*, Proceedings of the Informatika a felsőoktatásban 2011, pp. 269-277, ISBN 978-963-473-461-1, Debreceni Egyetem, Debrecen, Hungary, 2011.08.24-26.
[33] Sergiy V. Kovalevskyy, *Artificial Intelligence as a Driver of Socio-Economic System Transformation in Ukraine*, Interdisciplinary Description of Complex Systems, Vol. 22, Issue 3, pp. 296-304, DOI: 10.7906/indecs.22.3.5, 2024.